本文只讨论Linux文件自动备份方案，不同的技术方案适用于不同的需求场景。

备份是安全的基础根本保障，技术方案在实现上应考虑安全性。

需求场景1：本地Linux服务器文件备份，一周一备。

技术方案一: scp

1、公私钥免密连接

2、crontab+scp：

`0 0 * * 0 scp -r root@192.168.204.130:/tmp/ /tmp`

在局域网里面，这应该是最简单的方案了。

技术方案二：rclone

Rclone 是一款的命令行工具，支持在不同对象存储、网盘间同步、上传、下载数据。可使用rclone定时自动备份到网盘。

技术方案三：备份服务器

采用samba、Git等服务，搭建一个只写入不允许删除的备份服务器，然后定期将文件推送写入备份服务器。注：在备份中可使用tar压缩文件。

技术方案四：文件共享

采用hfs、nfs等服务，搭建文件共享，定期进行备份，这里就不详述啦。

需求场景2：将云上一台Linux服务器文件备份到本地服务器，一周一备即可。

首先，需要有一方开放端口服务，这里，我们将云服务器作为服务端，同时设置白名单只允许本地服务器出口IP才允许访问。

方案一：FTP

说起文件传输备份，首先想到就是FTP，FTP是用于网络上进行文件传输的一套标准协议，但使它声名狼藉的问题是它以明文方式传输密码和文件内容，只要在网络中对FTP连接进行监控就能被窃取。

FTP协议存在一些难以改善的缺点，它将走向终点。很显然，这并不是一个好的技术方案。

方案二：SFTP

云服务器作为服务端开启SFTP，提供连接地址、用户名、密码，白名单限制访问来源IP。

客户端可根据操作系统类型，采用不同的技术措施定期下载备份。

A、下载到Window服务器：

定时任务+WinSCP

```
winscp.exe /console /command "option batch continue" "option confirm off" "open sftp://username:password@192.168.204.130:22" "option transfer binary" "get /tmp D:\data\" "exit" /log=log_file.txt
```

B、下载到Linux服务器：

crontab+lftp

```
lftp -u username,password sftp://192.168.204.130 << EOF cd /tmp mget *.* bye EOF
```

方案三：rsync

1、云服务器搭建rsync服务端，设置账户密码，白名单限制访问来源IP。

2、本地服务器安装rsync客户端，编写shell脚本，结合crontab实现定时增量备份。

```
rsync -avz --password-file=/root/passwd root@192.168.204.130::common /tmp >/dev/null 2>&1
```

本文为小密圈《应急响应笔记》的小伙伴一起讨论的结果，Bypass负责整理发布，诚邀你一起加入分享知识。

应急响应笔记

星主： Bypass

知识星球

微信扫描预览星球详情